Journal of Machine Learning Research 1 (2003) 1-48 Submitted 4/00; Published 10/00

Filtering in Hybrid Dynamic Bayesian Networks

Morten Nonboe Andersen MNA@IMM.DTU.DK
Rasmus Ørum Andersen
Informatics and Mathematical Modelling (IMM)
Technical University of Denmark, Building 321
DK-2800 Lyngby, Denmark

Kevin Wheeler KWHEELER@MAIL.ARC.NASA.GOV
Computational Sciences Division
NASA Ames Research Center
Moffett Field, CA 94035-1000, USA

Editor:

Abstract

We implement a 2-time slice dynamic Bayesian network (2T-DBN) framework and make a 1-D state estimation simulation, an extension of the experiment in (v.d. Merwe et al., 2000) and compare different filtering techniques. Furthermore, we demonstrate experimentally that inference in a complex hybrid DBN is possible by simulating fault detection in a watertank system, an extension of the experiment in (Koller & Lerner, 2000) using a hybrid 2T-DBN. In both experiments, we perform approximate inference using standard filtering techniques, Monte Carlo methods and combinations of these. In the watertank simulation, we also demonstrate the use of 'non-strict' Rao-Blackwellisation. We show that the unscented Kalman filter and UKF in a particle filtering framework outperform the generic particle filter, the extended Kalman filter and EKF in a particle filtering framework with respect to accuracy in terms of estimation RMSE and sensitivity with respect to choice of network structure. Especially we demonstrate the superiority of UKF in a PF framework when our beliefs of how data was generated are wrong. Furthermore, we investigate the influence of data noise in the watertank simulation using UKF and PFUKD and show that the algorithms are more sensitive to changes in the measurement noise level that the process noise level. Theory and implementation is based on (v.d. Merwe et al., 2000).

Keywords: Hybrid Bayesian Networks, Dynamic Bayesian Networks, Particle Filtering, Extended Kalman Filter, Unscented Kalman Filter, Extended Kalman Particle Filter, Unscented Filter, Rao-Blackwellisation

1. Introduction

Bayesian networks (BN) (J. Pearl, 1988) have been used in a variety of problem domains, for example medical expert systems. However, often the problems treated in literature are very simple. Much of the work has been focused on static BN's with discrete, stochastic variables and linear variable relations where the model structure allows effective exact inference (Doucet et al., 1998). Models allowing time varying relations between network variables[1], *dynamic Bayesian networks* (DBN) (Dean & Kanazawa, 1989), allow for much

[1] Note that the network *structure* is not changed over time

more complex modelling. However, in discrete DBN's the cost of inference is exponential and in most hybrid (discrete and continuous valued) systems (Kalman filters being the only notable exception) the complexity of the belief state grows unboundedly over time (Lerner & Parr, 2001). The solution was the introduction of sequential Monte Carlo methods also known as particle filters (PF). In (Koller & Lerner, 2000) a PF has been applied to a discrete valued traffic monitoring DBN. One of the major drawbacks of PF, however, is that sampling in high dimensional spaces can be very difficult. In some cases, we can take advantage of the model structure and analytically marginalize out substructure(s) conditioned on the remaining (sampled) nodes using standard algorithms such as the Kalman filter. Hence, we can reduce the sampling space dramatically. This idea has been applied to a variety of problems such as concurrent localization and map learning for a mobile robot (Murphy & Russell, 2001) which operates in a discrete valued domain. In the hybrid domain, the technique has been applied to for example real-time monitoring of complex industrial processes (Morales-Menendez et al., 2002) and fault diagnostics (de Freitas, 2002), but using only *linear* variable relations. In many real-life problems, we need hybrid models that allow non-linear relations. Thus we need approximate inference techniques such as the extended Kalman filter (EKF) which is an estimator based on the Taylor series expansion of the non-linear functions (Grewal & Angus, 1993) or the unscented Kalman filter (UKF) (Julier & Uhlmann, 1997). The UKF is a recursive estimator that uses the true (non-linear) models and makes a Gaussian approximation of the distribution of the state random variable. Using EKF as a proposal generator in a PF framework leads to the extended Kalman particle filter Doucet et al. (1998) and using the UKF as proposal generator leads to the unscented Filter (v.d. Merwe et al., 2000) in this work abbreviated PFEKF and PFUKF resp. These techniques have been used in (v.d. Merwe et al., 2000) on a 1-D continuous valued model.

Working on non-linear, hybrid DBN's we allow modelling of many real-life problems. Hence, it is of great interest to investigate inference in these models. It is the aim of this paper to work on a real-life problem with linear as well as non-linear relations and keep all observations in their 'true' domain leading to models with both discrete and continuous valued nodes. We assume a Markovian, stationary model and setup a 2T-DBN in which nodes given at time t is dependent only on variables at time t and $t-1$. We compare several different techniques for inference and describe our observations in details. Initially, we compare the techniques on a dynamic 1-D continuous state estimation problem. The network is comparable with the one in (v.d. Merwe et al., 2000), but we use variable relations *different* from the ones used in the data generation to resemble the most common (and more difficult) situation that we do not know how exactly how data was generated. We also extend our model to include a third order relation. We visualize the strength of the UKF based implementations when the variable relations are of higher order and when the true variable relations are unknown. Next, we implement the non-linear, hybrid fault diagnostics DBN from (Koller & Lerner, 2000). However, we do not limit ourselves to inference using the generic PF (as in Koller & Lerner, 2000), but take advantage of the network structure. Analytically marginalizing out substructure(s) conditioned on the remaining (sampled) nodes is known as Rao-Blackwellisation (RB) (Casella & Robert, 1996). Combining RB with PF is known as Rao-Blackwellised particle filtering (RBPF). Strict RB is not possible with non-linear equations. In this experiment we apply a technique that we have named 'non-strict' Rao-Blackwellisation. That is we keep the non-linear relations, sample the discrete nodes and apply *approximate* inference to the remaining (Kalman) substructure as proposed as future work in (Koller & Lerner, 2000). We show that UKF and PFUKF are superior with respect to estimation root mean-square-error (RMSE) and discrete failure tracking compared to the generic PF, EKF and PFEKF. The performance of the generic PF, EKF and PFEKF are also shown to be highly sensitive to the choice of network structure. Hence, we propose a new network that solves this problem. An experiment with different levels of data noise shows that this aspect is also important for our modelling and choice of filtering algorithm. Next, we present an experiment using a measurement model different from the one used to generate the data as we do not want algorithms that fail when we do not know the true variable relations. We demonstrate the superiority of UKF in a PF framework when our beliefs of how data was generated are wrong. In (Koller & Lerner, 2000) no experiments are presented using different levels of noise or a false measurement model.

The paper is outlined as follows: First, we give a very brief overview of Bayesian networks, PF, EKF and UKF and the combination of these. Next, we present our 1-D simulation using a false process model. Finally, we present the more complex fault diagnostics simulation showing significant advantages using UKF and PFUKF.

2. Filtering in Bayesian Networks

In this paper, bold face symbols indicate a vector or a matrix and standard face symbols are scalars. We focus on experimental results based on existing theory. For a thorough description, please refer to (v.d. Merwe et al., 2000).

Converting a multivariate Gaussian distribution into a Bayesian network by ordering the variables $x_1, ..., x_n$ topologically (parents before children), the distribution of a child conditioned on its parents is computed as $P(x_i|x_1, ..., x_{i-1}) = \mathcal{N}\left(x_i; \beta_{i,0} + \sum_{j=1}^{i-1} \beta_{i,j}x_j, \sigma_i^2\right)$. An edge from x_j to $x_i (1 \leq j < i)$ corresponds to $\beta_{i,j} \neq 0$. Estimating the state of a system using a set of observations that becomes available on-line, i.e. filtering, is solved by modelling the evolution of the system. The general state space model (without control input) consists of a *state transition or state process* model and *a state measurement* model

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \qquad p(\boldsymbol{y}_t|\boldsymbol{x}_t) \tag{1}$$

where $\boldsymbol{x}_t \in \Re^{n_x}$ are the states (hidden variables) of the system at time t, $\boldsymbol{y}_t \in \Re^{n_y}$ are the observations and $p(\boldsymbol{x}_0)$ is the prior distribution at time $t = 0$. The state transitions are a first order Markov process and the observations are assumed to be independent given the states. For example, non-linear, non-Gaussian models can be expressed as

$$\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_{t-1}, \boldsymbol{v}_{t-1}), \qquad \boldsymbol{y}_t = \boldsymbol{h}(\boldsymbol{x}_t, \boldsymbol{n}_t) \tag{2}$$

where $\boldsymbol{v}_t \in \Re^{n_v}$ is the process noise and $\boldsymbol{n}_t \in \Re^{n_n}$ the measurement noise. From eq. (2) we get $p(\bar{\boldsymbol{x}}_t|\bar{\boldsymbol{x}}_{t-1}, \bar{\boldsymbol{v}}_{t-1}) = \delta\left(\bar{\boldsymbol{x}}_t - \boldsymbol{f}(\bar{\boldsymbol{x}}_{t-1}, \bar{\boldsymbol{v}}_{t-1})\right)$ and $p(\bar{\boldsymbol{y}}_t|\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{n}}_t) = \delta\left(\bar{\boldsymbol{y}}_t - \boldsymbol{h}(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{n}}_{t-1})\right)$ and obtain an expression for eq. (1): $p(\bar{\boldsymbol{x}}_t|\bar{\boldsymbol{x}}_{t-1}) = \int \delta\left(\bar{\boldsymbol{x}}_t - \boldsymbol{f}(\bar{\boldsymbol{x}}_{t-1}, \bar{\boldsymbol{v}}_{t-1})\right) p(\bar{\boldsymbol{v}}_{t-1}) d\bar{\boldsymbol{v}}_{t-1}$ and $p(\bar{\boldsymbol{y}}_t|\bar{\boldsymbol{x}}_t) = \int \delta\left(\bar{\boldsymbol{y}}_t - \boldsymbol{h}(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{n}}_t)\right) p(\bar{\boldsymbol{n}}_t) d\bar{\boldsymbol{n}}_t$. Our goal is to compute the filtering density $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ recursively to avoid computing the complete posterior density $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$. Thus we avoid keeping track of the complete history of the states and are still able to compute estimates of the mean, confidence intervals etc. of the system states.

In the extended Kalman filter the standard Kalman filter (for linear systems) is applied to non-linear systems with additive white noise by continually updating a linearization around the previous state estimate, starting with an initial guess. Hence, it is a minimum mean-square-error (MMSE) estimator based on the Taylor series expansion of the non-linear functions $\mathbf{f}$ and $\mathbf{h}$ around the estimates $\bar{\mathbf{x}}_{t|t-1}$ of the states $\mathbf{x}_t$, for example $\boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{v}_t) = \boldsymbol{f}(\bar{\boldsymbol{x}}_{t|t-1}, \bar{\boldsymbol{v}}_{t|t-1}) + \frac{\partial \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{v}_t)}{\partial \boldsymbol{x}_t}\Big|_{(\boldsymbol{x}_t = \bar{\boldsymbol{x}}_{t|t-1})}(\boldsymbol{x}_t - \bar{\boldsymbol{x}}_{t|t-1}) + \frac{\partial \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{v}_t)}{\partial \boldsymbol{v}_t}\Big|_{(\boldsymbol{v}_t = \bar{\boldsymbol{v}}_{t|t-1})}(\boldsymbol{v}_t - \bar{\boldsymbol{v}}_{t|t-1}) + \ldots$ As the EKF only uses the first order terms of the Taylor series expansion of the non-linear functions, it may introduce significant errors in the estimation of the posterior distribution of the states. Especially if the models are highly non-linear where the local linearity assumptions do not hold. The unscented Kalman filter is a recursive MMSE estimator that does not approximate the non-linear process and measurement models, but uses the true models and approximates the distribution of the state random variable. The state distribution is still represented by a Gaussian random variable, but by a *minimal* set of deterministically chosen sample points that completely capture the true mean and covariance of the Gaussian random variable. When this variable is propagated through the true non-linear system, it captures the true mean and covariance to the second order for any non-linearity. To calculate the statistics of a random variable undergoing a non-linear transformation, as required by the UKF, the scaled unscented transformation (SUT) (Julier & Uhlmann, 2002) is used. From a computational perspective, the UKF is superior to EKF, as it does not require explicit calculation of Jacobians (or Hessians), but computes a covariance matrix square root which can be done using a Cholesky factorization in order $n_x^3/6$. However, by expressing the covariance matrices recursively, this can be done in order n_x^2 using a recursive update to the Cholesky factorization (v.d. Merwe et al., 2000).

EKF and UKF both rely on a Gaussian approximation. PF represents a generalization of Monte Carlo methods for a dynamic process. It constructs the conditional probability of the state variables, with respect to the measurements, through a random exploration of the state space by entities called particles drawn from a *proposal distribution* (which should resemble the true posterior distribution as much as possible). A weight is assigned to each particle by a Bayes correction term based on the measurements. The algorithm approximates the initial distribution $P_0(\boldsymbol{x})$ by N Dirac measures (the particles). Each one has an initial probability of $1/N$ (the weight) and proportionally more particles are placed on the more probable regions of state space. This representation is equivalent to the bootstrap technique. Then the process evolution is done by particle propagation according to the model dynamics and, finally, the measurements are incorporated into the particles by the Bayes correction factor given by the conditional probability $P(\boldsymbol{y}|\boldsymbol{x}_i)$, where $\boldsymbol{y}$ is the measurement and $\boldsymbol{x}_i$ is the i-th particle state. Therefore, at each time step, the ensemble of particles estimates the a posteriori distribution of the state $\boldsymbol{x}$. In this paper our goal is to perform filtering on the given models, that is to compute a sequential estimate of the posterior distribution at time t without modifying the previously simulated states $\mathbf{x}_{0:t-1}$ allowing proposal distributions of the form $q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})$ Assuming the states follow a first order Markov process and that the observations are conditionally independent given the states yields $p(\mathbf{x}_{0:t}) = p(\mathbf{x}_0)\prod_{j=1}^{t} p(\mathbf{x}_j|\mathbf{x}_{j-1})$ and $p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}) = \prod_{j=1}^{t} p(\mathbf{y}_j|\mathbf{x}_j)$ and a recursive estimate for the importance weights $\omega_t = \omega_{t-1}\frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})}$. Now, given a proposal distribution and a set of prior samples, we are able to sequentially sample and evaluate likelihood, transition probabilities and importance weights leading to estimates of the state mean and covariance. The transition prior $q(\mathbf{x}_t|\mathbf{x}_{0:t-1},\mathbf{y}_{1:t}) \circeq p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the most popular choice of proposal distribution (even though it gives higher variance as it does not include the most recent observations) simply because it is easier to implement. We refer to this implementation as the generic PF. For an additive Gaussian process noise model the transition prior simplifies to

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}\left(f(\boldsymbol{x}_{t-1},0),\boldsymbol{Q}_{t-1}\right) \tag{3}$$

In practice, what happens is that after a few time steps one of the normalized importance weights is close to 1 while the rest is close to 0. In other words, a lot of the samples become useless and are neglected. To avoid this phenomenon, *degeneracy*, the particles need to be resampled (selection step) to eliminate particles with low importance weights and multiply particles with high importance weights. We implemented sampling importance resampling (SIR), multinomial sampling and residual resampling, see (v.d. Merwe et al., 2000) and (Liu et al., 2000) for details. They are all $O(N)$ algorithms. In the experiments, however, the choice of resampling algorithm did not seem to influence the results significantly and residual resampling was chosen arbitrarily. Another way to introduce sample variety after the selection step without affecting the validity of the approximation is to introduce a MCMC step of invariant distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ on each particle. If the particles are distributed according to the posterior $p(\tilde{\mathbf{x}}_{0:t}|\mathbf{y}_{1:t})$, and we apply a Markov chain transition kernel $\mathcal{K}(\mathbf{x}_{0:t}|\tilde{\mathbf{x}}_{0:t})$ with invariant distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ s.t. $\int \mathcal{K}(\mathbf{x}_{0:t}|\tilde{\mathbf{x}}_{0:t})\ p(\tilde{\mathbf{x}}_{0:t}|\mathbf{y}_{1:t}) = p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, we still have a set of particles distributed according to the posterior. However, the particles might have been moved to areas of higher likelihood and the total variance of the current distribution with respect to the invariant distribution can only decrease (Doucet et al., 2000b). In the generic PF this is implemented by sampling from the transition prior and accept according to a Metropolis-Hastings (MH) step. We refer to this filter as PFMC. Using the transition prior as proposal distribution we do not include the latest observation. To overcome this problem, we use the EKF and UKF as proposal generators.

In this work, the EKF approximates the optimal MMSE estimator of the system state. It computes the conditional mean of the state given all observations recursively through time by propagating the Gaussian approximation of the posterior distribution and combining it with the new observation available at each time step. That is, the EKF computes the recursive approximation of the true posterior filtering density given by

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx p_{\mathcal{N}}(\mathbf{x}_t|\mathbf{y}_{1:t}) = \mathcal{N}\left(\bar{\mathbf{x}}_t, \hat{\mathbf{P}}_t\right) \tag{4}$$

Using the EKF in a particle filtering framework, a separate EKF is used to generate and propagate a Gaussian proposal distribution for each particle $q(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)},\mathbf{y}_{1:t}) \doteq \mathcal{N}(\mathbf{x}_t|\mathbf{y}_{1:t})$, $i = 1,\ldots,N$. Thus, we need to propagate the covariance $\hat{P}^{(i)}$ and specify the EKF process and measurement noise covariances. Secondly, the i-th particle is sampled from this distribution. This filter is called the *extended Kalman particle filter*. However, even though the EKF moves the prior towards the likelihood, we are still faced with the Gaussian assumption on the form of the posterior and linearization approximations. Comparing eq. (4) with the Gaussian transition prior in eq. (3), it is noted that the proposal distribution generated by the EKF includes the most current observation at time t. In general though, the true form of this density will *not* be Gaussian - even with Gaussian process and measurement noise - which can be shown using a Bayes rule expansion of the proposal distribution. This implies that we are left with an experimental judgement of the gain versus the loss of filter performance. As mentioned, the UKF propagates the mean and covariance of the Gaussian approximation to the state distribution more accurately than the EKF and tends to generate better estimates of the true covariance of the state. Furthermore, the distributions generated by the UKF generally have a broader overlap with the true posterior distribution compared to the EKF estimates which is partly due to the fact that the UKF computes the posterior covariance accurately to the second order whereas the EKF uses a first order biased approximation. The UKF also includes the latest observations, but in a more accurate way. Hence, the UKF is more likely to generate more accurate proposal distributions within the particle filtering framework. Using the UKF as proposal distribution generator leads to the *unscented filter.*

3. 1-D simulation

Initially, we work on a 1-D continuous valued state estimation problem. We apply the generic PF and PFMC and compare these solutions with the EKF and the UKF. We also implement PFEKF and PFUKF with and without an MH step (referred to as PFEKFMC and PFUKFMC resp. using an MH step). For a more detailed presentation and more experiments using this network, please refer to (Andersen & Andersen, 2003). Our DBN has three nodes and is illustrated in Figure 1.

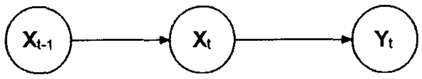

Figure 1: DBN for one-dimensional problem

The true process model is a noisy AR(1) model given by $x_t = 2 + cos(\alpha\pi t) + \beta x_{t-1}$. The measurement model is given by

$$y_t = \begin{cases} \theta_1 x_t & t <= 20 \\ \theta_2 {x_t}^2 & 20 < t <= 40 \\ \theta_3 {x_t}^3 & t > 40 \end{cases} \tag{5}$$

The measurement model was chosen to compare the algorithms for distributions with increasing non-linearity. In all experiments we use $\alpha = 0.05, \beta = 0.5, \theta_1 = -1, \theta_2 = 0.05$ and $\theta_3 = 0.01$ in the data generation. All experimental results were based on an average over 10 runs and all particle filters (all algorithms except EKF and UKF) used a fairly low number of particles (500) allowing us to show how the different particle filters perform without using too many samples. To simulate a real-life situation where the a prior knowledge of the system is often incomplete, we propose a process model different from the true one to simulate an approximate model. In (v.d. Merwe et al., 2000) experiments using a similar network are presented, but most importantly, experiments are only performed using the *true* process and measurement models. Furthermore, the measurement model in (v.d. Merwe et al., 2000) does not include a third order relation. The proposed (false) process model was $x_t = 2 + cos(\alpha\pi t) + \tilde{\beta}x_{t-1}$ using $\alpha = 0.05$ and $\tilde{\beta} = 0.2$. In the data generation we draw process and measurement noise samples from the Gaussian distributions $\mathcal{N}(0, 1)$ and $\mathcal{N}(0, 1e - 4)$ resp. and propose similar noise levels (that is we apply a reasonable process noise compared to our data values (in the range of approximately $0 - 12$) and a low measurement noise indicating reliable observations. The results of this experiment are shown in Table 1 and the tracking plots in Figure 2 and 3. In Figure 4 the noise level and the RMS error for the EKF/UKF and PFEKF/PFUKF state mean estimates (for $t = 41 - 60$) are shown.

Algorithm	RMSE	
	mean	var
Extended Kalman Filter (EKF)	0.8120	0
Unscented Kalman Filter (UKF)	0.3682	0
Particle Filter - generic (PF)	0.9666	8.4e-3
Particle Filter - Metropolis-Hastings move (PFMC)	0.9640	1.6e-3
Particle Filter - EKF proposal (PFEKF)	0.7977	3.9e-6
Particle Filter - EKF proposal and MH move (PFEKFMC)	0.7961	3.0e-5
Particle Filter - UKF proposal (PFUKF)	0.0823	1.4e-3
Particle Filter - UKF proposal and MH move (PFUKFMC)	0.0821	1.1e-3

Table 1: Mean and variance of RMSE values of state mean estimates using the false process and true measurement models, 500 particles and averaging over 10 runs. The variance is due to the 10 independent runs which include random steps.

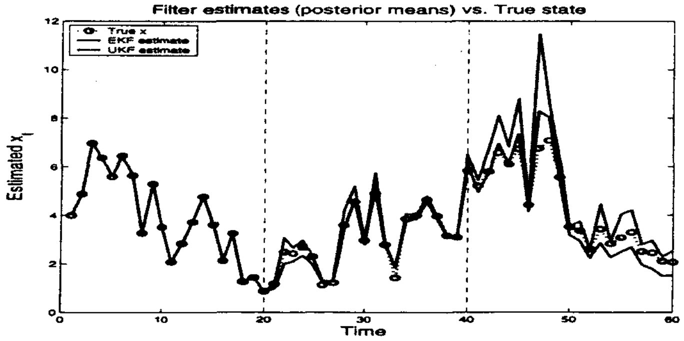

Figure 2: True state values, EKF and UKF estimates using the false process and true measurement models. Notice how EKF and UKF are similar in the first order stage, but in the second and third order stage the UKF estimates are consistently closer to the true state values than the EKF estimates.

The use of a false process model naturally makes it a lot harder for the algorithms to estimate the states as our beliefs of how the data was generated are no longer true. As

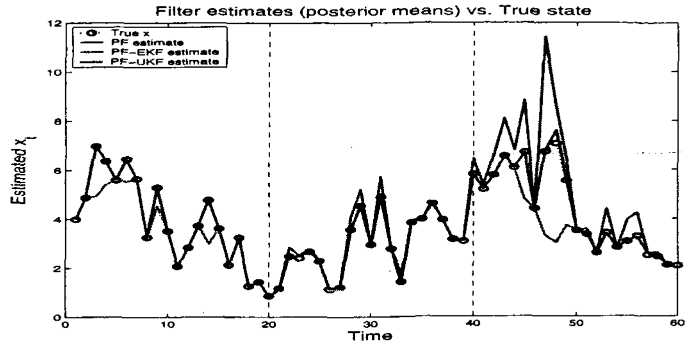

Figure 3: True state values, PF, PFEKF and PFUKF estimates using the false process and true measurement models Notice how EKF and UKF are similar in the first order stage where PF performs poorly in comparison, but in the second and third order stage the UKF estimates are consistently closer to the true state values than both the PF and the EKF estimates. The larger covariance estimates generated by UKF compared to EKF makes UKF better suited as proposal generator in a PF framework. PFUKF thus takes advantage of the good properties in UKF as well as in PF.

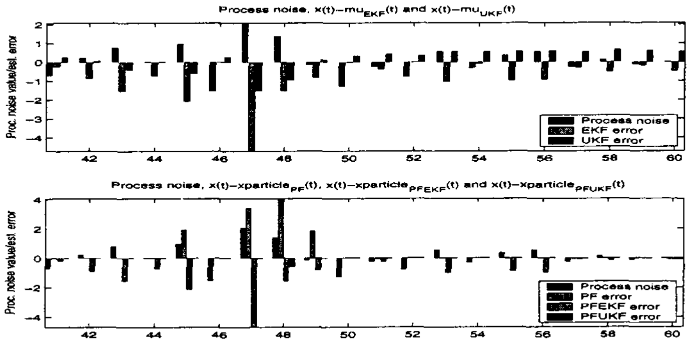

Figure 4: Noise level and RMS error for PF, PFEKF and PFUKF state mean estimates for $t > 40$ using the false process and true measurement models. Notice how the EKF estimates are consistently too large (negative bar values) and how PFEKF is not able to correct these errors. In comparison, UKF most often underestimates, but is much closer to the true state values than PF and PFEKF especially when the noise samples are large. PFUKF improves the UKF estimates very much and hardly makes any mistakes. In comparison, PF itself has a hard time when the noise samples are large as with PFEKF, but this simpler approach is actually better than PFEKF for small noise samples.

mentioned, this is the most likely situation in real life and thus we need a filter which does not fail dramatically when our beliefs are wrong. We notice how the generic PF is not capable of making better state mean estimates than EKF and UKF as sampling from the prior using a false process model leads to rather poor results. And the MH move is not enough to improve the situation significantly as the prior is too far from the likelihood. In similar experiments (data not shown) using the true process and measurement models, we observed that the MH step had a more positive effect on the results of the generic PF. However, it is interesting to see how the main loss of accuracy compared to EKF is in the first order stage of the measurement model which is seen by comparing the estimates of EKF in Figure 2 with the estimates of PF in Figure 3. This is where EKF is able to capture the true posterior mean and covariance. In the higher order stages PF is actually giving better state mean estimates than EKF. PFEKF/PFEKFMC perform slightly better than EKF taking advantage of the sampling as the state variance estimates of EKF are now partly based on the Jacobian of a false process model. It is still capable of tracking in the first order stage though as illustrated in Figure 2, but its limitations are shown in the higher order stages. And finally, PFUKF and PFUKFMC prove their superiority by keeping the estimation error on a very low level. UKF itself is suffering from the use of a false process model in the higher order stages, but is still capable of working as a proposal generator in a PF setup and perform better than the EKF based implementations. Similar experiments were performed using the true models and applying Gaussian and Gamma noise (data not shown). These experiments supported the results presented here, but the difference in performance was less significant as the problem was less complicated. Hence, we prefer PFUKF when we work on higher order models or feel uncertain that our beliefs are correct and have the necessary computational time. Otherwise, we would settle for UKF or the generic PF which has the advantage that we can work on model that do not have a Kalman structure.

The proposed (false) process model was $x_t = 2 + cos(\alpha \pi t) + \tilde{\beta} x_{t-1}$ using $\alpha = 0.05$ and $\tilde{\beta} = 0.2$.

4. Watertank simulation

Next, we apply our filtering techniques to a complex hybrid model presented in (Koller & Lerner, 2000) which models a process commonly used as a benchmark in the fault diagnostics (Mosterman & Biswas, 1999). The system is shown in Figure 5. For a more detailed presentation and more experiments using this network, please refer to (Andersen & Andersen, 2003). In the system $Tank1$ and $Tank2$ have pressure P_1 and P_2 resp. The watertanks

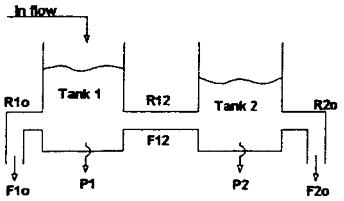

Figure 5: Illustration of the watertank system

are connected via a pipe with resistance R_{12} and flow F_{12}. Each tank has a pipe from which water flows out of the tank with resistances R_{1o} and R_{2o} and flows F_{1o} and F_{2o} resp. The first tank also has a constant flow of water F_{in} going into it. We get measurements $M1o$, $M12$ and $M2o$ of the flows in each of the three pipes. The 2T-DBN in Figure 6 describes how flow, pressure and resistance are related in theory. Practically, the process and measurement models and the measurements themselves are noisy. Furthermore, there are three possible type of failures in the system: **Measurement failure**: Usually measurements are quite reliable, but in the case of a measurement failure, the measurement becomes extremely noisy. **Pipe bursts**: A pipe can suddenly burst and change its resistance to some unknown value **Drifts**: The resistance of the pipe can drift gradually increaseing or decreasing the pipes resistance.

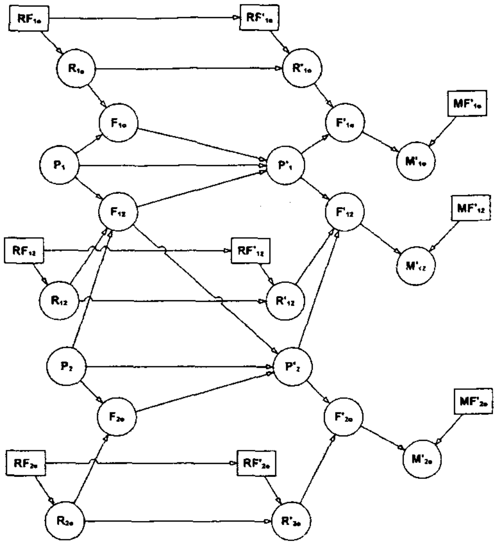

Figure 6: DBN for the watertank system

The 2T-DBN modelling the two-tank system is shown in Figure 6 with discrete variables depicted as rectangles and continuous variables circles. *RF* indicate pipe resistance faults (drifts or bursts) and *MF* indicate measurement failures. P, F and R are continuous valued and indicate pressure, flow and pipe resistance resp. M indicate observable flow measurements. All other variables are hidden. By explicitly modelling the pipe resistance, we accurately model the physical system. However, as the flow is the ratio between the pressure and the resistance we have to deal with ratios which is difficult, especially when the values are close to zero. Instead of modelling the resistances we choose to model the conductances (reciprocal of the resistance). This transformation results in products rather than ratios. The system is still non-linear, but this does not pose a problem for our algorithms. Note the difference between the measurement failure nodes and the pipe faults. The pipe faults are persistent and therefore appear both at time t and time $t+1$. Measurement failures, on the other hand, are transient and therefore appear only at time $t+1$ and need not be included in the belief state. As a result, the network has six pipe fault variables and three measurement failure variables, leading to 32,768 different discrete states. To simplify, the pipe connecting the two tanks can not burst. This would imply writing a new set of mass balance equations to model the physical system. As a consequence, the discrete variable state space is reduced to 18,432 states. Unfortunately, this network is still far too complicated to be able to use exact inference. In fact, the belief state at time t is a mixture of Gaussian's with a number of mixture components that grows exponentially with t, and with the number of discrete variables in each time slice.

Suboptimally, we would like to sample all discrete indicator variables, that is conductance and measurement failures, which can be grouped into two vector-valued nodes $\boldsymbol{CF}_t$ and $\boldsymbol{MF}_t$ and apply exact inference on the remaining continuous valued nodes, which we group into a single vector-valued node, $\boldsymbol{X}_t$. The observation nodes are likewise grouped into a single vector-valued node $\boldsymbol{Y}_t$ allowing a transformation of the fairly complicated network into a simple network as illustrated in Figure 7. The network corresponds exactly to the original network as all connections are expressed in form of the transition matrices, that is $\boldsymbol{CF}_t = \boldsymbol{A} \cdot \boldsymbol{CF}_{t-1}$, $\boldsymbol{X}_t = \boldsymbol{B}(\boldsymbol{CF}_t) \cdot \boldsymbol{X}_{t-1}$ and $\boldsymbol{Y}_t = \boldsymbol{D} \cdot \boldsymbol{X}_t$. This transformation allows us to sample part of the network and apply exact inference algorithms to the remaining part. A technique known as *Rao-Blackwellisation* (RB). Unfortunately, although the noise is Gaussian, the dynamics are non-linear, making it hard to integrate out $\boldsymbol{X}_t$. Hence, we apply our approximate inference techniques EKF and UKF and call it 'non-strict' RB. To compare, we also apply a generic PF, PFMC, PFEKF and PFUKF. 6 programs were created in Matlab implementing the 6 algorithms mentioned above and named accordingly. All programs except the generic PF and PFMC were designed as a two-step serial process. The first process samples the discrete nodes using a generic PF algorithm, but without updating the continuous state variables. The continuous states are then estimated (for each particle) in the second process using EKF, UKF, PFEKF or PFUKF. This two-step process was used as all the EKF and UKF based algorithms were able to give good estimates of the continuous nodes based on poor estimates of the discrete nodes due to the correction step in EKF and UKF. In the proposed network structure, the flow nodes are the only nodes directly connected to the observation nodes (see Figure 6). This design favors a good estimation of the flow using EKF and the generic PF. In EKF, the Kalman gain is partly based on the Jacobian of the measurement model. As the Jacobian is calculated by taking the derivative of the measurement model with respect to the 8 continuous state variables, only the flow variables will be represented in the Jacobian. Even though pressure and conductance are highly correlated with the flow, the Kalman gain only influences the flow estimates. As pressure at time t is calculated using pressure and flow at time $t-1$, a good flow estimate corresponds to a good pressure estimate. However, this is only valid if the pressure nodes are initialized correctly and if there is no noise. The conductance at

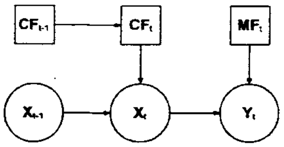

Figure 7: Simplified DBN for the watertank problem

time t is only based on the conductance at time $t - 1$ and the conductance failure at time t. Assuming the right conductance failures are chosen in the PF step, EKF will produce fairly good estimates of the conductance if it is initialized well and if the level of noise is low. Figure 8 shows the relative RMSE of the flow, conductance and pressure estimates for a typical run using EKF with correct initialization of all state variables. Both process and measurement noise samples were drawn from a Gaussian distribution $\mathcal{N}(0, 0.1)$ which is a reasonable noise level compared to the drifting factor of 2 used in this experiment. When the conductance (resistance) of a pipe changes, it is due to drifting (or a burst). To be able to distinguish between noise and drifting, the drifting factor should be fairly large compared to the noise values. Figure 8 illustrates that EKF is making poor conductance and pressure estimates whereas the flow estimates are very accurate for all time steps.

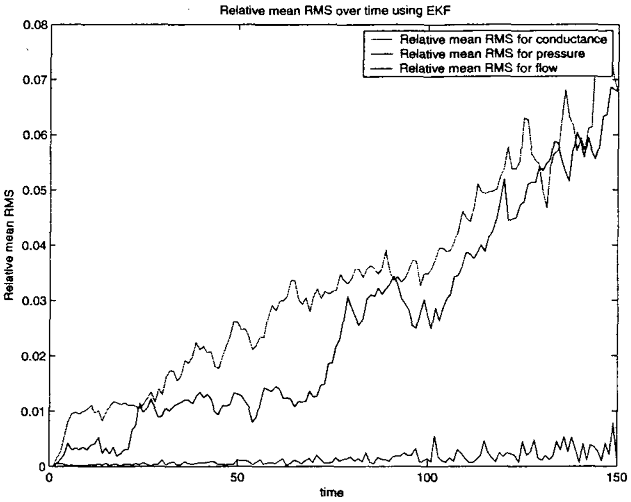

Figure 8: Relative RMSE for conductance (red graph), pressure (black graph) and flow (blue graph) estimates for a typical run using EKF with correct initialization. Notice how EKF is only capable of estimating the flow values.

In the generic PF, a number of particles are produced by sampling from the transition prior. In each time step, all particles are weighted according to their likelihood, that is based on the difference between the true and predicted values of the observation nodes. With only the flow nodes connected to the observation nodes, a particle with accurate flow values estimates will give a high likelihood regardless of whether the particle has poor conductance or pressure estimates. This problem is illustrated in figure 9 which shows how 10 particles are weighted for a given time step using the generic PF. The actual weights used follow the weights based on the flow values and not the optimal weights. A large process noise would make this problem even worse. In UKF, the Kalman gain is based on a number of sigma points that are propagated trough the network using the *true* process and measurement models. Both pressure and conductance are highly correlated with the observation nodes, even though they are not directly connected, which makes UKF able to

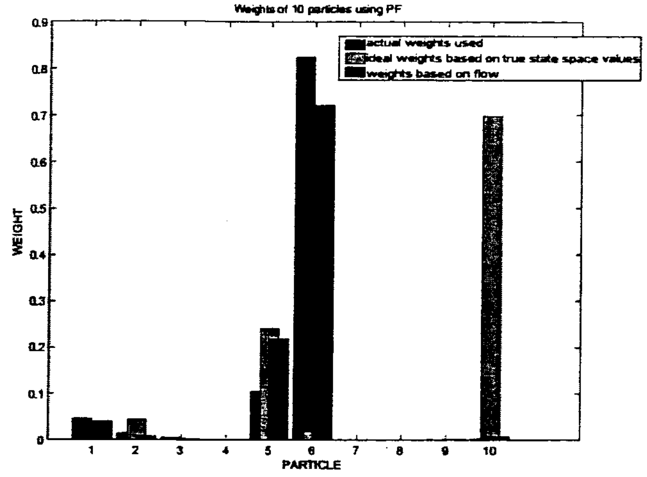

Figure 9: The actual weights used in the generic PF (blue bars), the optimal weights based on the distance from the true continuous state values (green bars) and the weights based on the distance to the true flow values (red bars) for 10 particles. Notice how the actual weights used follow the weights based on the flow values and not the optimal weights.

updates all continuous state variables. One of the objectives in the watertank problem is to track failures making accurate estimation of the conductance a crucial point. Hence, a new network was proposed by eliminating the flow nodes allowing conductance and pressure nodes to be directly connected to the observation nodes. The flow measurements are now predicted based on pressure and conductance plus noise. When data is generated using the old network, noise is added to the flow nodes making the data more noisy in the new network. To compare the two networks, the old network was used to generate datasets to be used in both networks. One would expect the old network to make more accurate estimates than the new network using a data set generated by itself. Figure 10 shows the average RMSE for the conductance and pressure estimates from the two networks using the generic PF, EKF and UKF. The results are based on 10 different data sets using 10 runs for each data set. Again, the process and measurement noises samples were drawn from a Gaussian distribution $\mathcal{N}(0, 0.1)$ and the conductance drifting set to 2 (change in resistance units per time unit). As illustrated, the new network outperforms the old network for all continuous state mean estimates using PF and EKF. The difference in performance for EKF becomes even more obvious using a poor initialization of the pressure and conductance values (data not shown). In comparison, the performance of UKF does not depend on the network structure. Based on the experiments, the new network structure was preferred and used in all forthcoming experiments. Based on the simulation in sec. 3 and the presented experiment, we see no further reason to include the EKF based implementations. We keep the generic PF algorithms for comparison.

To evaluate our estimates of the discrete failure nodes we need to compare the process

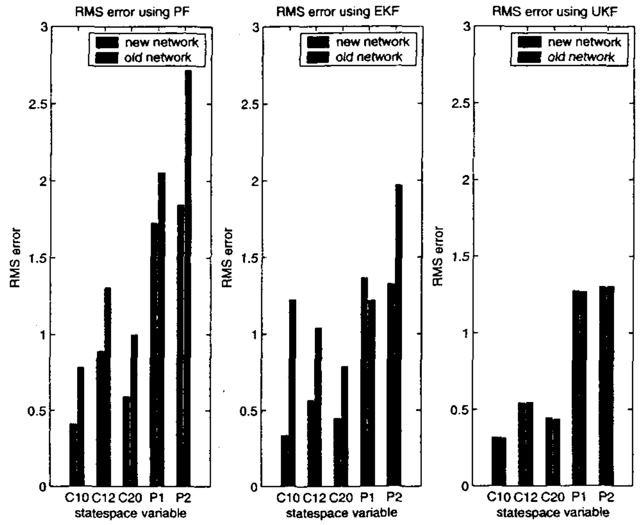

Figure 10: Average RMSE for conductance and pressure estimates using the old network (red bars) and the new network (blue bars) with PF, EKF and UKF resp. The performance of UKF algorithm is independent of the network structure as opposed to PF and EKF.

and measurement noise levels with the extent to which a certain failure influences the flow. To give an indication of how robust the algorithms are with respect to the noise level, experiments were performed changing the level of noise in the data. Only UKF and PFUKF were used in this experiment as they showed superior performance in the previous experiments. In all experiments the true noise levels were used as process and measurement noise proposals in the filtering algorithms. Four different process and measurement noise levels were used, $\mathcal{N}(0, \sigma^2), \sigma^2 = 0.01, 0.1, 0.2$ and 0.4. The continuous state mean estimation RMSE and the number of wrong failure estimates as a function of the noise levels are shown in Figure 11. A time period of 100 time steps were used and the results based on 20 different data sets using 10 runs for each data set. Outliers were removed. Notice the nice correlation between the RMSE and the number of wrong failure estimates for UKF (two left plots in Figure 11) and PFUKF (two right plots in Figure 11). An accurate continuous state estimate corresponds to a small RMSE making it easier to track the failures and vice versa. Both the RMSE and the average number of wrong failure estimates using UKF and PFUKF are more influenced by the level of measurement noise than the process noise as the state estimates are updated *based on* the estimates of the observation nodes. A noisy connection between state variables and observation variables is thus more severe than a noisy process model. Furthermore, Figure 11 once again illustrates the relationship between UKF and PFUKF. PFUKF is doing much better than UKF for large measurement noise levels, but notice that the RMSE increases using the smallest process noise level (0.01). Here, PFUKF is actually doing worse than UKF. When UKF makes accurate estimates, PFUKF can make matters

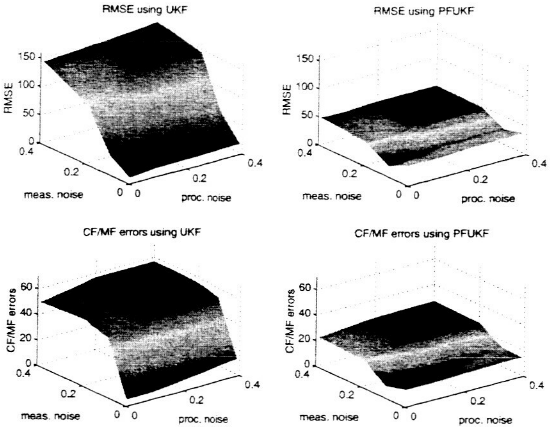

Figure 11: Surface plots showing the RMSE (top plots) and CF/MF estimation errors (lower plots) using UKF (left plots) and PFUKF (right plots) based on 16 different measurement and process noise combinations

worse by either fitting the measurement noise or sampling from a Gaussian distribution that is too wide.

In a real life simulation one of the major challenges is to come up with reasonable process and measurement models. In a simulation we know the true models and hence UKF is able to make very accurate state estimates leaving no real space for improvement by further sampling in PFUKF. Hence, PFUKF should prove its superiority in scenarios where UKF is not given optimal conditions. To investigate this, we experiment by changing the proposed measurement model by simply adding 5% to all the flow estimates. PF and PFMC were also included in the experiments for the sake of comparison. 100 particles and 30 subparticles (particles used in the estimation of the continuous states) were used over a time period of 60 time steps. The process and measurement noise samples were drawn from the Gaussian distribution $\mathcal{N}(0, 0.1)$. Table 2 shows the mean continuous state mean estimation RMSE and variance using 10 different data sets and 10 runs for each data set. The second column shows the average number of incorrect failure estimates. PFUKF does not surprisingly show the best ability to deal with a wrong measurement model as this was already indicated in the 1-D simulation in sec. 3. PFUKF is able to move the particles towards regions of higher likelihood reducing the RMSE and making tracking of the discrete failure nodes easier. We might be able to further improve the PFUKF estimates using more particles or subparticles. The experiment once again showed that PFUKF is capable of making more accurate state estimates than UKF when the algorithms are not given conditions. Furthermore, UKF and PFUKF made more accurate state estimates than PF and PFMC. The latter algorithms both fail when our beliefs of how data was generated are wrong as indicated in the experiment in sec. 3.

In (2000), Koller & Lerner apply a generic PF to the watertank problem using the network structure in Figure 6. They propose as future work to use a generic PF to sample the discrete failure nodes and a more sophisticated PF to sample the continuous variables as

Algorithm	RMS		CF/MF errors
	mean	var	
Particle Filter - generic (PF)	256	204	56.3
Particle Filter - Metropolis-Hastings move (PFMC)	227	187	33.8
Unscented Kalman Filter (UKF)	208	43	30.1
Particle Filter - UKF proposal (PFUKF)	178	76	23.4

Table 2: Mean and variance of RMSE values of state estimates and average conductance and fault detection errors using a wrong measurement model.

in this paper. It is impossible to make direct comparisons between this work and the work of Koller and Lerner as pipe bursts etc. can be modelled differently. However, as we also use the generic PF, a comparison between this and the other filtering techniques can be made. In sec. 4 we showed that the generic PF (and EKF) are highly sensitive to the choice of network structure and that the UKF based implementations were superior with respect to continuous state mean estimation RMSE and discrete failure tracking. UKF and PFUKF have outperformed both the generic PF and PFMC in every single experiment. In our experiments we were able to track all continuous variable and discrete failure combinations very well using UKF or PFUKF and a very low number of particles. Figure 12 shows the tracking of $C10$ (conductance of pipe between $Tank1$ and the outside world and the different events occurring during a typical simulation, similar results were obtained for the other continuous variables - data not shown) for a simulation of 100 time steps using UKF. We present a tracking plot for UKF instead of the superior PFUKF to show that we can track the continuous variables and detect system faults very well using a very low number of particles compared to the 50000 particles used in the generic PF in Koller & Lerner (2000) without taking advantage of PFUKF, which is computationally more expensive than UKF. A drifting factor of 1 was used and both process and measurement noise samples were drawn from the Gaussian distribution $\mathcal{N}(0, 0.5)$. These settings gave a low SNR allowing us to illustrate that tracking is possible even in a very noisy environment. The results were based on ten runs with one data set using 300 particles. Notice the large error bar at time step 31 in Figure 12 corresponding to the burst of pipe 3 at time step 30 (only every second error bar plotted for visual reasons). Apparently, UKF had no problem recognizing the measurement failures as these events are not seen in the error bars for any of the three variables, as this would have caused very large errorbars at the given points in time. Furthermore, notice the larger error bars in Figure 12 when the drifting ends at $t = 52$ which might indicate that the system in some runs estimated positive/negative drift of $C10$ as this is not crucial for the flow measurements, when the conductance is very high (low resistance). At $t = 80$ it locates the burst of pipe 1 and stays at the bursting level. As mentioned, in (Koller & Lerner, 2000), 50000 particles were used in the generic PF. We have shown that the new network structure is better suited for PF (and EKF) than the original structure and that the UKF based implementations still outperformed the generic PF using using a very low number of particles.

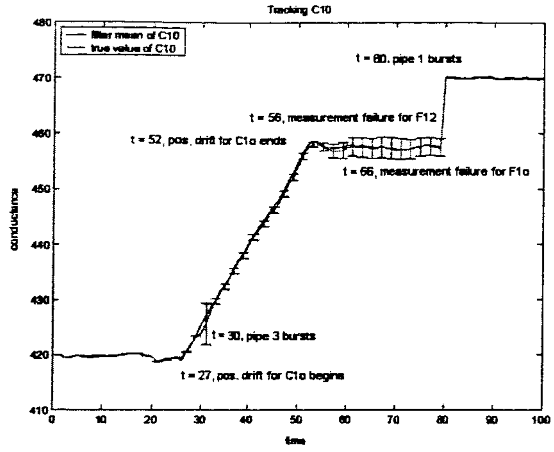

Figure 12: Tracking of conductance variable $C10$. The estimated conductance C10 (red line) using UKF is plotted with confidence intervals (plus, minus two standard deviations from the mean estimate) and the true conductance C10 (black line).

5. Conclusion

In this paper we have implemented an extended Kalman filter, an unscented Kalman filter, a generic particle filter, a generic particle filter with MCMC steps, a particle filter with EKF proposal and MCMC steps, a particle filter with UKF proposal and MCMC steps and applied the different filters to a 2T-DBN in a simulation of a continuous valued 1-D state estimation problem comparable with the experiment presented in (v.d. Merwe et al., 2000). However, our measurement model was extended to include a third order stage. We experimented using the true process and measurement models with Gaussian and Gamma noise added (data not shown) and an experiment using a process model different from the true one. All experiments indicated that PFUKF (PFUKFMC) were the most accurate and reliable filters. The latter setup was the most important experiment as most real-life applications involve approximations to the 'true' (unknown) process and measurement models.

Next, we applied 6 of the 8 filters to a hybrid 2T-DBN simulation of a watertank problem presented in (Koller & Lerner, 2000). We compared two network designs and showed that PF and EKF were able to make good estimates of those continuous state variables that were connected directly to the observation nodes. However, the estimates of the remaining variables were very inaccurate. In comparison, UKF was able to update all state space variables regardless of their connection to the observation nodes and UKF thus performed equally well in both networks. The original network structure was discarded and the EKF based algorithms not used in further experiments.

Then we experimented with different levels of data noise. The continuous state mean estimates using UKF and PFUKF were influenced more by the measurement noise level than the level of process noise. Large measurement noise levels made the UKF estimates poor and PFUKF was able to move the particles towards the true state, hence reducing the RMSE.

Finally, we experimented using a measurement model different from the true one by adding 5% to all the flow estimates. Again, PFUKF was capable of making more accurate state estimates than UKF when the algorithms were not given 'optimal' (true models and low level of noise) conditions. Furthermore, UKF and PFUKF were making more accurate state estimates than the generic PF and PFMC. Section 4 showed that we were able to track the discrete failure nodes with a fairly low number of particles using UKF. These results are to some extent comparable with the work of Koller and Lerner in (Koller & Lerner, 2000) in which a generic PF was applied to the watertank problem using the network structure in Figure 6. We have shown that a different network structure merely improved the accuracy of the generic PF and still the use of UKF or PFUKF significantly improved the ability to track the true failure nodes and estimate the continuous state variables with a low number of samples.

All in all, we have compared several inference techniques and shown that it is possible to do inference in a complex hybrid DBN. These networks allow for many complex real-life problems to be modelled. We conclude that we should choose PFUKF when we work on higher order models, when the measurements are noise (that is when UKF is not able to make reliable estimates), when we do not know the 'true' process and measurement models and have the necessary computational time. Otherwise, we would settle for UKF or standard PF. However, PF has the disadvantage that is is network structure dependent.

Acknowledgements

We would like to thank the funds/companies that have supported this work financially: Civilingeniør Frants Allings fond, Vera og Carl Johan Michaelsens fond, Julie Damms Studiefond, Alexandre Haynman og hustru Nina Haynmans fond, Incentive Fonden, Ingeniørforeningen i Danmark (IDA), Familien Hede Nielsens Fond, Danmarks Tekniske Universitet, RS Components A/S, Oticon Fonden, Knud Højgaards Fond.

References

M.N. Andersen and R.Ø. Andersen. *Filtering in Hybrid Dynamic Bayesian Networks* Master thesis, IMM, Techinical University of Denmark, 2003.

G. Casella and C.P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika,* 83(1): 81-94, 1996.

T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Artificial Intelligence* 93(1-2): 1-27, 1989.

A. Doucet, S. Godsill and C. Andrieu. On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. Technical report CUED/F-INFENG/TR 310, Dept. of Eng., Cambridge University, 1998.

A. Doucet, N. de Freitas, K. Murphy and S. Russell. *A Simple Tutorial on RBPF for DBNs.* 2000.

N. de Freitas. Rao-Blackwellised Particle Filtering for Fault Diagnosis. *IEEE Aerospace,* pages 176-183, 2002.

S.M. Grewal and P.A. Angus. Kalman Filtering Theory and Practice. Prentice Hall, Upper Saddle River, NJ, 1993.

S.J. Julier and J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Proceedings of AeroSense: The 11'th International Symposiom on Aerospace/Defence Sensing, Simulation and Controls*, Multi Sensor Fusion, Tracking and Resource Management II. Orlando, Florida, 1997.

S.J. Julier and J.K. Uhlmann. The Scaled Unscented Transformation. In *Proceedings of the IEEE American Control Conference*, pages 4555-4559, 2002

D. Koller and U. Lerner. *Sampling in Factored Dynamic Systems.* In Sequential Monte Carlo Methods in Practice: pages 445-464, Springer-Verlag, New York, 2000.

U. Lerner and R. Parr. Inference in Hybrid Networks: Theoretical Limits and Practical Algorithms. In *Proceedings of the 17'th Annual Conference on Uncertainty in AI (UAI)*, pages 310-318, 2001.

J. Liu, R. Chen and T. Logvinenko. A theoretical framework for sequential importance sampling with resampling. Technical report, Dept. of Stat., Stanford University, California, 2000.

Ruben Morales-Menendez, Nando de Freitas and David Poole. *Real-Time Monitoring of Complex Industrial Processes with Particle Filters.* NIPS 2002.

R. v.d. Merwe, A. Doucet, N. de Freitas and E. Wan. The Unscented Particle Filter. Technical report CUED/F-INFENG/TR-380, Dept. of Eng., Cambridge University, 2000.

P.J. Mosterman and G. Biswas. *Diagnosis of Continuous Valued Systems in Transient Operating Regions* Knowledge Systems Laboratory, November, 1999.

K. Murphy and S. Russell. A. Doucet, N. de Freitas and N.J. Gordon, editors. *Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks.* In Sequential Monte Carlo Methods in Practice, Springer-Verlag, New York, 2001.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems*, Morgan-Kaufmann, San Francisco, 1988.